

DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL

```
IIIIII  MM  MM  AAAAAA  GGGGGGGG  EEEEEEEEE  XX  XX  EEEEEEEEE  CCCCCCCC  TTTTTTTTTT
IIIIII  MM  MM  AAAAAA  GGGGGGGG  EEEEEEEEE  XX  XX  EEEEEEEEE  CCCCCCCC  TTTTTTTTTT
II  MM  MM  AA  AA  GG  GG  EE  XX  XX  EE  CC  TT
II  MM  MM  AA  AA  GG  GG  EE  XX  XX  EE  CC  TT
II  MM  MM  AA  AA  GG  GG  EE  XX  XX  EE  CC  TT
II  MM  MM  AA  AA  GG  GG  EE  XX  XX  EE  CC  TT
II  MM  MM  AA  AA  GG  GG  EE  XX  XX  EE  CC  TT
II  MM  MM  AA  AA  GG  GG  EE  XX  XX  EE  CC  TT
II  MM  MM  AA  AA  GG  GG  EE  XX  XX  EE  CC  TT
II  MM  MM  AA  AA  GG  GG  EE  XX  XX  EE  CC  TT
II  MM  MM  AA  AA  GG  GG  EE  XX  XX  EE  CC  TT
II  MM  MM  AA  AA  GG  GG  EE  XX  XX  EE  CC  TT
IIIIII  MM  MM  AA  AA  GGGGGG  EEEEEEEEE  XX  XX  EEEEEEEEE  CCCCCCCC  TTTTTTTTTT
IIIIII  MM  MM  AA  AA  GGGGGG  EEEEEEEEE  XX  XX  EEEEEEEEE  CCCCCCCC  TTTTTTTTTT
```

```
LL  IIIIII  SSSSSSSS
LL  IIIIII  SSSSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SSSSSS
LL  II  SSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```

(3)	115	RUN IMAGE
(4)	152	LOGOUT PROCESS
(5)	192	PREPARE THE PROCESS FOR LOGOUT
(6)	237	MCR COMMAND
(7)	281	EXTERNAL COMMAND EXECUTION
(8)	323	INITIATE IMAGE
(9)	547	DCL\$FORCEEXIT - FORCE IMAGE EXIT
(10)	574	ALLOCATE BUFFER AND BUILD DESCRIPTOR
(11)	602	COMMAND INTERPRETER EXIT HANDLER

```

0000 1 .TITLE IMAGEXECT - IMAGE EXECUTION
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 ALL RIGHTS RESERVED.
0000 10
0000 11 THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 TRANSFERRED.
0000 17
0000 18 THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 CORPORATION.
0000 21
0000 22 DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26
0000 27
0000 28 IMAGE EXECUTION BY EXTERNAL IMAGE COMMAND OR RUN COMMAND
0000 29
0000 30 D. N. CUTLER 4-APR-77
0000 31
0000 32 MODIFIED BY:
0000 33
0000 34 V03-008 HWS0090 Harold Schultz 22-Jul-1984
0000 35 Just delete the process if an image activation error
0000 36 is encountered while attempting to logout via executing
0000 37 the LOGINOUT image.
0000 38
0000 39 V03-007 HWS0075 Harold Schultz 28-Jun-1984
0000 40 Move initialization of command interpreter command
0000 41 pointers to the beginning of the command interpreter
0000 42 exit handler in order to insure this initialization in
0000 43 the event a CNTL-Y is pending
0000 44
0000 45 V03-006 HWS0037 Harold Schultz 21-Mar-1984
0000 46 Use PRC_V_IRUNDWN flag to indicate whether or not an
0000 47 image has been rundown by DCL
0000 48
0000 49 V03-005 MSH0001 Michael S. Harvey 12-Jan-1984
0000 50 Defer reporting alternate success message until image
0000 51 fixups are completed. This is necessary because message
0000 52 output can clobber fixup context stored in P1.
0000 53
0000 54 V03-004 PCG0008 Peter George 27-May-1983
0000 55 Change DCL$RESTORE_OUTPUT calling format.
0000 56
0000 57 V03-003 PCG0007 Peter George 13-Jan-1983

```



IMAGEXECT  
V04-000

- IMAGE EXECUTION

D 13

15-SEP-1984 23:54:20 VAX/VMS Macro V04-00  
4-SEP-1984 23:41:04 [DCL.SRC]IMAGEXECT.MAR;1

Page 2  
(1)

0000 58 :  
0000 59 :  
0000 60 :  
0000 61 :  
0000 62 :  
0000 63 :  
0000 64 :  
0000 65 :  
0000 66 :  
0000 67 :---

Set up exit handler block in INITIAL, not here.  
Close redefined SYSS\$OUTPUT before logging out.  
Add DCL\$CLOSE\_PPFS.

V03-002 PCG0006 Peter George 22-Oct-1982  
Move DCL\$EOJ to COMMAND.

V03-001 PCG0005 Peter George 09-Jun-1982  
Allow MCR command/qualifier without a delimiting blank.

```

0000 69 : MACRO LIBRARY CALLS
0000 70 :
0000 71 :
0000 72 :
0000 73 $PPDDEF ;PROCESS PERMANENT DATA AREA
0000 74 PRCDEF ;DEFINE PROCESS WORK AREA
0000 75 WRKDEF ;DEFINE COMMAND WORK AREA
0000 76 PTRDEF ;RESULT PARSE DESCRIPTORS
0000 77 $CLMSGDEF ;DEFINE ERROR/STATUS VALUES
0000 78 $CHFDEF ;DEFINE CONDITION ARGLIST OFFSETS
0000 79 $FABDEF ;DEFINE FAB OFFSETS
0000 80 $IFDDEF ;IMAGE FILE DESCRIPTOR DEFINITIONS
0000 81 $IHDDEF ;IMAGE HEADER DESCRIPTOR DEFINITIONS
0000 82 $NAMDEF ;DEFINE NAME BLOCK OFFSETS
0000 83 $PSLDEF ;DEFINE PROCESSOR STATUS FIELDS
0000 84 $RABDEF ;DEFINE RAB OFFSETS
0000 85 $STSDEF ;DEFINE STATUS LONG WORD VALUES
0000 86 $CLIDEF ;DEFINE IMAGE ARGUMENT LIST FORMAT
0000 87 :
0000 88 :
0000 89 : LOCAL DATA
0000 90 :
0000 91 :
00000000 92 .PSECT DCL$ZCODE,BYTE,RD,NOWRT
3A 4D 45 54 53 59 53 24 53 59 53 00' 0000 93 EXTDEFAULT: ;EXTERNAL COMMAND IMAGE DEFAULT STRING
45 58 45 2E 000C 94 .ASCIC 'SYS$SYSTEM:.EXE'
OF 0000
0010 95 RUNDEFAULT: ;RUN COMMAND IMAGE DEFAULT STRING
45 58 45 2E 00' 0010 96 .ASCIC '.EXE'
04 0010
0015 97 LOGOUTIMG: ;FILENAME TEXT STRING FOR LOGOUT IMAGE
54 55 4F 4E 49 47 4F 4C 00' 0015 98 .ASCIC 'LOGINOUT'
08 0015
001E 99 MCRIMG: ;IMAGE FOR MCR COMMAND
58 53 52 00' 001E 100 .ASCIC 'RSX'
03 001E
0022 101 SYSPRINT: ;DEFAULT QUEUE LOGICAL NAME
54 4E 49 52 50 24 53 59 53 0022 102 .ASCII 'SYS$PRINT'
00000009 002B 103 SYSPRTSIZ = . - SYSPRINT
002B 104 ;
002B 105 :
002B 106 : DEFINE IMAGE NAMES
002B 107 :
002B 108 .MACRO INTIMAGE NAME
002B 109 IMG K 'NAME = $INTIMAGES
002B 110 $INTIMAGES = $INTIMAGES + 1
002B 111 .ENDM
002B 112
002B 113 INTIMAGES

```

```

002B 115 .SBTTL RUN IMAGE
002B 116 :+
002B 117 DCL$RUN - RUN IMAGE
002B 118 :
002B 119 THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE RUN COMMAND.
002B 120 :
002B 121 INPUTS:
002B 122 :
002B 123 R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
002B 124 R9 = ADDRESS OF SCRATCH STACK.
002B 125 R10 = BASE ADDRESS OF COMMAND WORK AREA.
002B 126 R11 = BASE ADDRESS OF PROCESS WORK AREA.
002B 127 :
002B 128 OUTPUTS:
002B 129 :
002B 130 THE SPECIFIED IMAGE IS INITIATED.
002B 131 :-
002B 132
002B 133 DCL$RUN::
0600 8F AA 002B 134 BICW #<PRC_M_DBGQUAL!PRC_M_DBGTRUE>, - ;RUN IMAGE
68 AB 002F 135 PRC_W_FLAGS(R11) ;AND DEBUG QUALIFIER TRUE FLAGS.
7E 7C 0031 136 -(SP) ;MAKE A SCRATCH BUFFER
FF CA' 30 0033 137 10$: BSBW DCL$GETDVAL ;GET NEXT DESCRIPTOR VALUES
04 55 91 0036 138 CMPB R5,#PTR_K_ENDLINE ;IS THIS THE END OF THE PARAMETERS?
1A 13 0039 139 BEQL 30$ ;BR IF YES
03 55 91 003B 140 CMPB R5,#PTR_K_PARAMETR ;IS THIS THE FILE TO RUN?
10 13 003E 141 BEQL 20$ ;BR IF YES
0600 8F AB 0040 142 BISW #<PRC_M_DBGQUAL!PRC_M_DBGTRUE>, - ;SET DEBUG QAULIFIER SEEN
68 AB 0044 143 PRC_W_FLAGS(R11) ;AND ASSUME QUALIFIER TRUE
EA 53 E9 0046 144 R3,T0$ ;BR IF NOT NEQATED
0049 145 PRC_V_DBGTRUE,PRC_W_FLAGS(R11) ;SET STATE TO FALSE
E3 11 004E 146 BRB 10$
6E 51 7D 0050 147 20$: MOVQ R1,(SP) ;SAVE FILE SPECIFICATION DESCRIPTOR
DE 11 0053 148 BRB 10$ ;LOOK FOR MORE
06 BA 0055 149 30$: POPR #*M<R1,R2> ;GET FILE SPEC OF FILE TO RUN
00A0 31 0057 150 BRW SETRUNDEF ;SET RUN DEFAULT AND GO RUN IT

```



```

005A 152      .SBTTL  LOGOUT PROCESS
005A 153      :+
005A 154      : DCL$LOGOUT - LOGOUT THE PROCESS
005A 155      :
005A 156      : THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE LOGOUT
005A 157      : COMMAND OR TO EFFECT A LOGOUT WHEN END OF FILE OCCURS FOR A DETACHED JOB.
005A 158      : ALSO, MAY BE ENTERED IF A HANGUP OCCURS ON A PROCESS WITH A DIAL UP INPUT.
005A 159      :
005A 160      : INPUTS:
005A 161      :
005A 162      :     R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
005A 163      :     R9 = ADDRESS OF SCRATCH STACK.
005A 164      :     R10 = BASE ADDRESS OF COMMAND WORK AREA.
005A 165      :     R11 = BASE ADDRESS OF PROCESS WORK AREA.
005A 166      :
005A 167      : OUTPUTS:
005A 168      :
005A 169      :     THE LOGOUT IMAGE IS INITIATED.
005A 170      : -
005A 171      :
005A 172      : DCL$LOGOUT::
005A 173      :
005A 174      : IF SILENT LOGOUT SPECIFIED, THEN SKIP LOGOUT AND BRANCH TO THERE.
005A 175      :
005A 176      :     BBC      #PRC V_AUTOLOGO,PRC_W_FLAGS(R11),10$ ; BRANCH IF FLAG NOT SET
005F 177      :     BRW      SILENT_LOGOUT
0062 178      :
0062 179      :
0062 180      : RUN DOWN ACTIVE PROCEDURES AND IMAGES.  CLOSE OPEN PPF FILES.
0062 181      :
0062 182      : 10$:  BSBB      DCL$CLOSE_PPFS
0064 183      :
0064 184      :
0064 185      : ACTIVATE LOGOUT IMAGE
0064 186      :
0064 187      :     MOVL      PRC_L_LSTSTATUS(R11), - ;PASS FINAL STATUS TO LOGOUT
0068 188      :     G^CTL$AG CLIDATA+PPD$L_LSTSTATUS
006D 189      :     MOVAB     LOGOUTIMG,R2 ;START OF THE COUNTED STRING
0071 190      :     BRB      SETIMGNAME ;GO SET LENGTH, DEFAULT, & EXECUTE IT

```

03 68 AB 08 E1  
FF9E' 31

OF 10

00B0 CB D0  
00000018'GF  
52 A5 AF 9E  
42 11



```

0073 192 .SBTTL PREPARE THE PROCESS FOR LOGOUT
0073 193 :+
0073 194 : DCL$CLOSE_PPFS - PREPARE THE PROCESS FOR LOGOUT
0073 195 :
0073 196 : THIS ROUTINE IS CALLED TO PREPARE THE PROCESS FOR LOGOUT. ACTIVE COMMAND
0073 197 : PROCEDURES AND IMAGES ARE RUN DOWN. OPEN PPF FILES ARE CLOSED.
0073 198 :
0073 199 : INPUTS:
0073 200 :
0073 201 : R11 = BASE ADDRESS OF PROCESS WORK AREA.
0073 202 :
0073 203 : OUTPUTS:
0073 204 :
0073 205 : NONE
0073 206 :-
0073 207
0073 208 DCL$CLOSE_PPFS::
0073 209 :
0073 210 : RUN DOWN ANY PROCEDURES OR IMAGES STILL ACTIVE
0073 211 :
0073 212 10$: TSTL PRC_L_INDEPTH(R11) ;INDIRECT LEVEL ZERO?
0076 213 BEQL 20$ ;IF EQL YES
0078 214 BSBW DCL$UNSTACK ;UNSTACK INDIRECT LEVEL
007B 215 BRB 10$ ;
007D 216
007D 217 :
007D 218 : CLOSE ALL PROCESS PERMANENT FILES STILL OPEN
007D 219 :
007D 220 20$: MOVL PRC_L_INDFAB(R11),R9 ;GET ADDRESS OF INDIRECT FAB
59 1C AB D0 0081 221 MOVAB PRC_L_PPFLIST(R11),R2 ;GET ADDRESS OF FILE DESCRIPTOR LISTHEAD
52 70 AB 9E 0085 222 30$: MOVL (R2),R2 ;GET ADDRESS OF NEXT FILE DESCRIPTOR
52 52 62 D0 0088 223 BEQL 40$ ;IF EQL END OF LIST
1C A2 B0 008A 224 MOVW RAB$L_CTX+4(R2),- ;INSERT INTERNAL FILE INDEX
02 A9 008D 225 FAB$W-IFI(R9)
008F 226 $CLOSE FAB=(R9) ;CLOSE FILE
EB 11 0098 227 BRB 30$ ;
009A 228
009A 229 :
009A 230 : CLOSE POSSIBLY REDEFINED SYSS$OUTPUT FILE
009A 231 :
52 0114 CB 9E 009A 232 40$: MOVAB PRC_W_OUTIFI(R11),R2 ;GET ADDRESS OF PROCESS PERMANENT OUTPUT FIL
58 00BC CB D0 009F 233 MOVL PRC_L_IDFLNK(R11),R8 ;GET ADDRESS OF CURRENT IDF FRAME
FF59' 30 00A4 234 BSBW DCL$RESTORE_OUTPUT ;CLOSE TEMPORARY OUTPUT FILE
05 00A7 235 RSB

```

```

00A8 237      .SBTTL  MCR COMMAND
00A8 238      :+
00A8 239      DCL$MCR - EXECUTE THE MCR COMMAND
00A8 240      :
00A8 241      THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE MCR COMMAND
00A8 242      :
00A8 243      INPUTS:
00A8 244      :
00A8 245      R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
00A8 246      R9 = ADDRESS OF SCRATCH STACK.
00A8 247      R10 = BASE ADDRESS OF COMMAND WORK AREA.
00A8 248      R11 = BASE ADDRESS OF PROCESS WORK AREA.
00A8 249      :
00A8 250      OUTPUTS:
00A8 251      :
00A8 252      THE FIRST TOKEN ON THE COMMAND LINE IS USED TO ACTIVATE AN IMAGE.
00A8 253      IF THE COMMAND LINE IS NULL, THE MCR IMAGE IS ACTIVATED.
00A8 254      :-
00A8 255      :
00A8 256      .ENABL  LSB
00A8 257      DCL$MCR:
00A8 258      BSBW      DCL$GETDVAL      :EXECUTE THE MCR COMMAND
00A8 259      MOVL      R1,R4             :GET DESCRIPTOR VALUE
00A8 260      BNEQ      10$              :COPY LENGTH OF STRING
00A8 261      MOVAB     MCRIMG,R2        :BR IF IMAGE NAME SUPPLIED
00A8 262      SETIMGNAME:                :SET ADDRESS OF IMAGE NAME
00A8 263      MOVZBL    (R2)+,R1         :SET IMAGE NAME, DEFAULT, & EXECUTE IT
00A8 264      BRB       SETIMGDEF        :SET IMAGE NAME LENGTH
00A8 265      10$:      MOVL      R1,R3    :GO SET DEFAULT AND EXECUTE IT
00A8 266      LOCC      #^A/ /,R3,(R2)   :SAVE LENGTH OF STRING
00A8 267      MOVQ      R0,R5             :FIND SPACE DELIMITER
00A8 268      LOCC      #^A/*/,R3,(R2)   :SAVE POSITION INDICATORS
00A8 269      CMPL      R1,R6             :FIND SLASH DELIMITER
00A8 270      BLSS      15$              :WHICH CAME FIRST?
00A8 271      MOVQ      R5,R0             :USE POSITON OF SLASH
00A8 272      15$:      SUBL      R0,R4    :USE POSITION OF SPACE
00A8 273      LOCC      #^A/[/,R4,(R2)   :FIND LENGTH OF TOKEN
00A8 274      BNEQ      20$              :LOOK FOR A DIRECTORY
00A8 275      LOCC      #^A/</,R4,(R2)   :BR IF DIRECTORY IN THE SPEC
00A8 276      BEQL      30$              :TRY OTHER SYNTAX
00A8 277      20$:      BBSS      #PRC_V_RUNDEF,PRC_B_FLAGS2(R11),30$ :BR IF NO DIRECTORY FOUND
00A8 278      30$:      MOVL      R4,RT   :USE RUN DEFAULT
00A8 279      .DSABL    LSB              :GET BYTE COUNT OF IMAGE TO RUN

```

```

00E9 281      .SBTTL  EXTERNAL COMMAND EXECUTION
00E9 282      :
00E9 283      :+ DCL$EXTIMAGE - EXTERNAL COMMAND EXECUTION
00E9 284      :
00E9 285      : THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO INITIATE EXECUTION OF AN
00E9 286      : EXTERNAL IMAGE.
00E9 287      :
00E9 288      : INPUTS:
00E9 289      :
00E9 290      :     R1 = LENGTH OF IMAGE FILE SPECIFICATION.
00E9 291      :     R2 = ADDRESS OF IMAGE FILE SPECIFICATION.
00E9 292      :     R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
00E9 293      :     R9 = ADDRESS OF SCRATCH STACK.
00E9 294      :     R10 = BASE ADDRESS OF COMMAND WORK AREA.
00E9 295      :     R11 = BASE ADDRESS OF PROCESS WORK AREA.
00E9 296      :
00E9 297      : OUTPUTS:
00E9 298      :
00E9 299      :     THE SPECIFIED IMAGE IS INITIATED.
00E9 300      :
00E9 301      :
00E9 302      DCL$EXTIMAGE::
00E9 303      SETIMGDEF:
00E9 304      BICW      #<PRC_M_DBGQUAL!PRC_M_DBGTRUE>, - ;EXTERNAL COMMAND EXECUTION
00ED 305      PRC_W_FLAGS(R11) ;SET CORRECT DEFAULT FOR IMAGE
00EF 306      MOVAB    EXTDEFAULT,R3 ;CLEAR DEBUG QUALIFIER SEEN,
00F4 307      BBCC    #PRC_V_RUNDEF,PRC_B_FLAGS2(R11),FORCEXIT ;AND DEBUG QUALIFIER TRUE FLAGS.
00FA 308      SETRUNDEF: ;SET IMAGE DEFAULT FOR EXTERNAL IMAGE
00FA 309      MOVAB    RUNDEFAULT,R3 ;SET RUN DEFAULT FOR IMAGE
00FF 310      : ;SET IMAGE DEFAULT FOR RUN'S
00FF 311      :
00FF 312      : FORCE IMAGE EXIT IF ONE IS ACTIVE
00FF 313      :
00FF 314      : INPUTS:
00FF 315      :
00FF 316      :     R1 = LENGTH OF IMAGE FILE SPECIFICATION
00FF 317      :     R2 = ADDRESS OF IMAGE FILE SPECIFICATION
00FF 318      :     R3 = ADDRESS OF COUNTED ASCII STRING FOR DEFAULT FILE NAME
00FF 319      :
00FF 320      :
00FF 321      :
023F 30 00FF 320      FORCEXIT:
00FF 321      BSBW    DCL$FORCEXIT ;FORCE IMAGE EXIT
00FF 321      :

```



```

0102 323 .SBTTL INITIATE IMAGE
0102 324
0102 325 INITIATE AN IMAGE BY ENTERING USER MODE AND THEN SETTING UP THE INITIAL CALL
0102 326 FRAMES AND ESTABLISHING A CATCH-ALL CONDITION HANDLER.
0102 327
0102 328 INPUTS:
0102 329
0102 330 R1 = LENGTH OF IMAGE FILE SPECIFICATION
0102 331 R2 = ADDRESS OF IMAGE FILE SPECIFICATION
0102 332 R3 = ADDRESS OF COUNTED ASCII STRING FOR DEFAULT FILE NAME
0102 333
0102 334
0102 335 .ENABL LSB
0102 336 INITIATE:
0102 337 BBSS #PRC_V_EXIT,PRC_W_FLAGS(R11),10$ ;INITIATE IMAGE
0102 338 PUSH R1 ;IF SET, EXIT HANDLER ESTABLISHED
0102 339 $DCLEXH S PRC_L_EXTBLK(R11) ;REMEMBER LENGTH OF IMAGE FILE SPEC
0102 340 MOVL -(SP)+,RT ;DECLARE EXIT HANDLER
0102 341 BLBS R0,10$ ;RESTORE LENGTH OF IMAGE FILE SPEC
0102 342 RSB ;BRANCH IF SUCCESSFUL
0102 343 ;RETURN IF ERROR
0102 344
0102 345 SET UP IMAGE ADDRESS SPACE
0102 346
0102 347
0102 348 10$: MOVAL G^MMG$IMGHDRBUF,R5 ;ADDRESS TO RETURN IMAGE HEADER INFO
0102 349 MOVQ R1,-(SP) ;PUT IMAGE NAME DSC ON STACK
0102 350 MOVL SP,R1 ;GET ADR OF FILNAM STRING DESCRIPTOR
0102 351 MOVZBL (R3)+,R2 ;GET SIZE OF DEFAULT NAME STRING
0102 352 MOVQ R2,-(SP) ;PUT DEFAULT NAME DSC ON STACK
0102 353 MOVL SP,R0 ;GET ADR OF DFLTNAM STRING DESCRIPTOR
0102 354 $IMGACT S - ;ACTIVATE THE IMAGE
0102 355 NAME=(R1),- ;ADDRESS OF FILE NAME DESCRIPTOR
0102 356 DFLNAM=(R0),- ;ADDRESS OF DEFAULT NAME DESCRIPTOR
0102 357 HDRBUF=(R5) ;ADDRESS OF IMAGE HEADER BUFFER
0102 358 BLBC R0,11$ ;BRANCH IF ACTIVATION ERROR
0102 359 ADDL #16,SP ;CLEAN OFF IMGACT DESCRIPTORS
0102 360 BRW 30$ ;AND GO ACTIVATE THE IMAGE
0102 361 11$: CLRQ (SP) ;SET UP NULL RESULT NAME DESCRIPTOR
0102 362 MOVL R0,R3 ;SAVE STATUS
0102 363 $RUNDOWN S #PSL$C_USER ;RUN DOWN IMAGE
0102 364
0102 365 CHECK IF THE IMAGE THAT JUST FAILED WAS LOGINOUT.
0102 366
0102 367 CMPB LOGOUTING,8(SP) ;CHECK LENGTH FIRST
0102 368 BNEQ 111$ ;BR IF NOT SAME LENGTH
0102 369 CMPC3 8(SP),@12(SP),LOGOUTING+1 ;WAS IMAGE LOGINOUT?
0102 370 BNEQ 111$ ;NO, PROCESS NORMAL ERROR PATH
0102 371
0102 372 FAILED IMAGE ACTIVATION WAS LOGINOUT. WILL USE $EXIT_S TO TERMINATE
0102 373 THE PROCESS INSTEAD
0102 374
0102 375 MOVL PRC_L_INDFAB(R11),R0 ;GET ADDR OF INDIRECT FAB
0102 376 MOVW PRC-W-OUTIFI(R11),- ;GET INTERNAL FILE INDEX OF SYS$OUTPUT
0102 377 FAB$W-IFI(R0)
0102 378 $CLOSE FAB=(R0) ;CLOSE INDIRECT OUTPUT FILE
0102 379 $CANEXH S ;CANCEL SUPERVISOR MODE EXIT HANDLERS

```

```

018B 380 $EXIT_S PRC_L_LSTSTATUS(R11) ;EXIT PROCESS WITH FINAL STATUS
0196 381
0196 382
55 08 54 D4 0196 383 111$: CLRL R4 ;INIT NULL STV FROM FAB
08 29 13 0198 384 MOVL 8(R5),R5 ;FAB ADDRESS OF ERROR IF ANY
52 2C 29 13 019C 385 BEQL 14$ ;BRANCH IF NONE PRESENT
08 34 23 13 019E 386 MOVL FAB$L_FNA(R5),R2 ;ORIGINAL FILE NAME ADDRESS
51 AE 51 9A 01A2 387 BEQL 14$ ;BRANCH IF FAB IS EMPTY
54 0C 51 7D 01A4 388 MOVZBL FAB$B_FNS(R5),R1 ;SIZE OF ORIGINAL FILE NAME STRING
50 28 51 7D 01A8 389 MOVQ R1,8(SP) ;USE THIS FOR ORIGINAL FILE NAME
52 04 A0 D0 01AC 390 MOVL FAB$L_STV(R5),R4 ;GET FAB'S STV FIELD
51 03 A0 D0 01B0 391 MOVL FAB$L_NAM(R5),R0 ;GET ADDRESS OF NAME BLOCK
06 12 01B4 392 MOVL NAM$L_RSA(R0),R2 ;ADDRESS OF NAME STRING
51 0B A0 9A 01B8 393 MOVZBL NAM$B_RSL(R0),R1 ;SIZE OF RESULT NAME STRING IF ANY
03 13 01BC 394 BNEQ 12$ ;BRANCH IF IT EXISTS
55 5E D0 01BE 395 MOVZBL NAM$B_ESL(R0),R1 ;EXPANDED NAME STRING SIZE IF ANY
6E 51 7D 01C2 396 BEQL 14$ ;BRANCH IF NO NAME TO OUTPUT
55 5E D0 01C4 397 MOVQ R1,(SP) ;RESULT/EXPANDED NAME STRING
01C7 398 14$: MOVL SP,R5 ;R5 IS ADR OF NAME DESCRIPTORS
01CA 399
01CA 400 0(SP) = DESCRIPTOR FOR RESULT/EXPANDED FILE NAME
01CA 401 8(SP) = DESCRIPTOR FOR ORIGINAL FILE NAME
01CA 402
52 000388AA 8F D0 01CA 403 MOVL #CLIS_IMGNAME,R2 ;ERROR CODE FOR 2ND MSG
53 00000000 8F D1 01D1 404 CMPL #RMS$_FNF,R3 ;IF ORIGINAL ERR IS FILE NOT FOUND
09 12 01D8 405 BNEQ 16$
53 53 D4 01DA 406 CLRL R3 ;THEN 3RD MSG IS NULL
52 000388B2 8F D0 01DC 407 MOVL #CLIS_IMAGEFNF,R2 ;AND 2ND IS "IMAGE FILE NOT FOUND"
01E3 408
01E3 409 : NOW BUILD THE PUTMSG ARGUMENT
01E3 410
01 53 0C 10 ED 01E3 411 16$: CMPZV #STSSV_FAC_NO,#STSS$_FAC_NO,R3,#1 ;IF THIS IS AN RMS ERROR CODE
02 12 01E8 412 BNEQ 17$
54 DD 01EA 413 PUSHL R4 ;THEN USE SAVED STV FROM FAB
53 DD 01EC 414 17$: PUSHL R3 ;ERROR CODE FOR 3RD MSG
65 D5 01EE 415 TSTL (R5) ;IF NO RESULT NAME STRING
09 12 01F0 416 BNEQ 18$
52 000388AA 8F D1 01F2 417 CMPL #CLIS_IMGNAME,R2 ;THEN SKIP THE IMAGE NAME MESSAGE
06 13 01F9 418 BEQL 182$
65 7F 01FB 419 18$: PUSHAQ (R5) ;ADR OF DESCRIPTOR FOR RSLT NAME
01 DD 01FD 420 PUSHL #1 ;ONE FAO ARGUMENT
52 DD 01FF 421 PUSHL R2 ;ERR CODE FOR 2ND MSG
08 A5 7F 0201 422 182$: PUSHAQ 8(R5) ;ADR OF DESCRIPTOR FOR ORIG NAME
01 DD 0204 423 PUSHL #1 ;ONE FAO ARGUMENT
00030000 8F DD 0206 424 PUSHL #<SHRS_ACTIMAGE ! - ;ERR CODE FOR 1ST MSG
020C 425 <CLIS_IMGNAME & STSSM_FAC_NO>
7E 55 5E C3 020C 426 SUBL3 SP,R5,-(SP) ;FORM SIZE OF ARG LIST IN BYTES
6E 04 C6 0210 427 DIVL #4,(SP) ;ARG LIST SIZE IN LONG WORDS
0213 428
0213 429 : NOW OUTPUT THE ERROR MESSAGES
0213 430
50 5E D0 0213 431 MOVL SP,R0 ;ADDRESS OF PUTMSG PARAMETERS
5E 55 FDE7 30 0216 432 BSBW DCL$PUTMSG ;CALL THE PUTMSG FACILITY
50 10 C1 0219 433 ADDL3 #16,R5,SP ;POP ALL INFO OFF STACK
50 53 D0 021D 434 MOVL R3,R0 ;GET ERROR CODE TO RETURN
03 12 0220 435 BNEQ 19$ ;BRANCH IF THIS IS THE ONE
50 52 D0 0222 436 MOVL R2,R0 ;ONLY 2 MESSAGES, USE THE 2ND

```

```

00 50 1C E2 0225 437 19$: BBSS #STSSV_INHIB_MSG,R0,20$ ;INHIBIT ERROR MESSAGE OUTPUT
05 0229 438 20$: RSB ;RETURN WITH STATUS CODE IN R0
022A 439
022A 440
022A 441 : RAISE ACCESS LEVEL TO USER
022A 442
022A 443
022A 444
022A 445 30$:
51 65 D0 022A 445 MOVL (R5),R1 ;GET ADDRESS OF IMAGE HEADER
50 00' D1 022D 446 CMPL S^#SS$ _NORMAL,R0 ;IF OTHER THAN NORMAL COMPLETION
13 13 0230 447 BEQL 35$
50 00000000'8F D1 0232 448 CMPL #SS$ _SYSVERDIF,R0 ;'SYSTEM VERSION MISMATCH'?
0A 12 0239 449 BNEQ 35$ ;BRANCH IF NOT
02 A1 28 B1 023B 450 CMPW #IHDS$ _SYSVER,IHDS$ _ACTIVOFF(R1) ;IF RELEASE 1 IMAGE
023F 451 ;WITH NO SYSVER STORED IN HEADER
04 19 023F 452 BLSS 35$ ;THEN AVOID ISSUING SPURIOUS WARNING
50 00'8F 9A 0241 453 MOVZBL #SS$ _NORMAL,R0 ;BY CONVERTING TO NORMAL STATUS
0245 454 35$:
01 E1 0245 455 BBC #IHDS$ _LNKNOTFR,- ;BRANCH IF TRANSFER ADDRESS PRESENT
19 20 A1 0247 456 IHDS$ _LNKFLAGS(R1),37$
50 00' D1 024A 457 CMPL S^#SS$ _NORMAL,R0 ;DID IMAGE ACTIVATION SUCCEED NORMALLY?
03 13 024D 458 BEQL 36$ ;IF EQL YES
FDAE' 30 024F 459 BSBW DCL$ERRORMSG ;PRINT ALTERNATE SUCCESS MESSAGE
0252 460 36$: $RUNDOWN S #PSL$C _USER ;RUN DOWN IMAGE IF NO XFER ADDRESS
50 0003889A 8F D0 025B 461 MOVL #CLIS _NOTFR,R0 ;AND RETURN STATUS OF 'NO TRANSFER ADDRESS'
05 0262 462 RSB
0080 CB 50 D0 0263 463 37$: MOVL R0,PRC _L _IMGACTSTS(R11) ;STORE IMGACT STATUS FOR LATER
50 04 A5 D0 0268 464 MOVL 4(R5),R0 ;IMAGE FILE DESCRIPTOR BLOCK ADDRESS
026C 465
026C 466 ASSUME IFDS$ _EXEONLY+1 EQ IFDS$ _PRIV
026C 467 ASSUME PRC _V _EXEONLY+1 EQ PRC _V _PRIV
50 10 A0 02 00 EF 026C 468 EXTZV #IFDS$ _EXEONLY,#2,IFDS$ _FLAGS(R0),R0 ;GET 'EXECUTE ONLY'
00AF CB 02 03 50 F0 0272 469 ;AND 'PRIVILEGED IMAGE BITS'
0272 470 INSV R0,#PRC _V _EXEONLY,#2,PRC _B _FLAGS2(R11) ;SAVE IN PROCESS FLAGS
F4 AA 5E D0 027D 471 SETBIT PRC _V _IRUNDWN,PRC _B _IMGF$ _AG(R11) ;INDICATE IMAGE ACTIVATED
7E OF 16 78 0281 472 MOVL SP,PRC _L _SAVSP(R10) ;SAVE CURRENT STACK POINTER
97'AF 9F 0285 474 ASHL #PSL$ _PRVMOD,#PSL$C _USER@2+PSL$C _USER,-(SP) ;SET USER PSL
0288 475 DCL$LOW _LIMIT:: B*50$ ;SET USER PC
09 68 AB 01 E1 0288 476 #PRC _V _CNTRLY,PRC _W _FLAGS(R11),40$ ;IF CLR, NO AST PENDING
7E 50 7D 028D 477 MOVQ R0,-(SP) ;PUSH R0 AND R1
7E 05 7D 0290 478 MOVQ #5,-(SP) ;PUSH #5 AND ZERO ASTPRM
FDA6' 31 0293 479 BRW DCL$SCNTRLY ;SIMULATE A CONTROL Y
02 0296 480 40$: REI ;ENTER USER MODE
0297 481 DCL$HIGH _LIMIT:: ;HIGH LIMIT OF CONTROL Y/C ADDRESS RANGE
0297 482
0297 483
0297 484 : BUILD TOP LEVEL CALL FRAME
0297 485
0297 486
0297 487 50$: CLRQ AP ;CLEAR INITIAL ARGUMENT AND FRAME POINTERS
9D'AF 5C 7C 0297 487 CALLS #0,B*60$ ;CONSTRUCT TOP LEVEL CALL FRAME
0299 488
029D 489
029D 490
029D 491 : ESTABLISH CATCH-ALL CONDITION HANDLER AND CALL IMAGE
029D 492
029D 493

```



6D	00000000'GF	0000	029D	494	60\$:	.WORD	0	:ENTRY MASK
		9E	029F	495		MOVAB	G^EXESCATCH_ALL,(FP)	:ESTABLISH CATCH-ALL HANDLER
			02A6	496		\$SETEXV	S #2,G^EXESCATCH_ALL	:ESTABLISH LAST CHANCE HANDLER
			02B9	497		\$IMGFIX	-S	:PERFORM ADDRESS RELOCATION
0080	CB	00'	D1	02C0	498	CMPL	S^SSS_NORMAL,PRC_L_IMGACTSTS(R11)	:NORMAL IMAGE ACTIVATION?
		2A	13	02C5	499	BEQL	65\$	:IF EQL YES
		5D	DD	02C7	500	PUSHL	RO	:SAVE IMGFIX STATUS
				02C9	501			
		00	DD	02C9	502	PUSHL	#0	:CREATE PUTMSG VECTOR (FAO COUNT)
0080	CB	DD	02CB	503		PUSHL	PRC_L_IMGACTSTS(R11)	:SET IMGACT STATUS CODE
		02	DD	02CF	504	PUSHL	#2	:# ARGS ON PUTMSG VECTOR
50	5E	DD	02D1	505		MOVL	SP,RO	:ADDRESS OF THE BUFFER DESCRIPTOR
				02D4	506			
204C4344	8F	DD	02D4	507		PUSHL	#^A'DCL '	: FACILITY NAME
		5E	DD	02DA	508	PUSHL	SP	: MAKE DESCRIPTOR OF NAME
		03	DD	02DC	509	PUSHL	#3	
		5E	DD	02DE	510	PUSHL	SP	: SET ADDRESS OF FACNAM
		00	DD	02E0	511	PUSHL	#0	: NO ACTION ROUTINE
		50	DD	02E2	512	PUSHL	RO	: RO = ADDRESS OF MESSAGE VECTOR
00000000'GF	06	FB	02E4	513		CALLS	#6,G^SYSS\$PUTMSG	: WRITE THE MESSAGE TO SYSS\$ERROR,OUTPUT
	5E	OC	CO	02EB	514	ADDL	#12,SP	: RESTORE THE STACK
				02EE	515			
50	8E	DD	02EE	516		MOVL	(SP)+,RO	:RESTORE IMGFIX STATUS
	47	50	E9	02F1	517	BLBC	RO,110\$	:IF LBC, WASH UP IMAGE
54	00000000'GF	7D	02F4	518	65\$:	MOVQ	G^MMG\$IMGHDRBUF,R4	:GET IMAGE HEADER DESCRIPTOR
				02FB	519			
				02FB	520	ASSUME	CLISV_DEBUG EQ 0	
				02FB	521	ASSUME	CLISV_DBGTRU EQ 1	
				02FB	522	ASSUME	PRC_V_DBGTRUE EQ PRC_V_DBGQUAL+1	
				02FB	523			
02	09	EF	02FB	524		EXTZV	#PRC V_DBGQUAL #2,-	:BUILD PROTOTYPE CLI OPTIONS VALUE BY
7E	68	AB	02FE	525			PRC Q_FLAGS(R11),-(SP)	:GETTING THE DEBUG QUALIFIER AND STATE
03	68	AB	0301	526		BBC	#PRC V_MODE,PRC W_FLAGS(R11),70\$	:BR IF NOT A BATCH JOB
			0306	527		SETBIT	CLISV_BATCH,(SP)	:OR IN THE BATCH BIT IF THIS IS BATCH
03	68	AB	0309	528	70\$:	BBC	#PRC V_VERIFY,PRC W_FLAGS(R11),80\$	:BR IF VERIFY IS CLEAR
			030E	529		SETBIT	CLISV_VERIFY,(SP)	:PROPOGATE VERIFY IF TRUE
5C	AB	D5	0311	530	80\$:	TSTL	PRC_L_INDEPTH(R11)	:INDIRECT LEVEL ZERO?
		03	13	0314	531	BEQL	90\$	:IF EQL YES
			0316	532		SETBIT	CLISV_INDIRECT,(SP)	:PASS INDIRECT NON ZERO FLAG
6E	18	93	0319	533	90\$:	BITB	#<CLISV_INDIRECT!CLISV_BATCH>,(SP)	:COMMANDS COMING FROM FILE?
			031C	534		BEQL	100\$	:BR IF TERMINAL JOB
03	6E	02	E1	031E	535	BBC	#CLISV_VERIFY,(SP),100\$	:BR IF VERIFY NOT REQUESTED
			0322	536		SETBIT	CLISV_VFYINP,(SP)	:INDICATE INPUT VERIFY IS NEEDED
20	A4	DD	0325	537	100\$:	PUSHL	IHDR_LNKFLAGS(R4)	:PASS LINKTIME OPTION FLAGS
7E	54	7D	0328	538		MOVQ	R4,-(SP)	:NEXT TWO PARAMETERS IN USER FRAME
0000'CF	9F	032B	539			PUSHAB	W^DCL\$UTLSERV	:SET ADDRESS OF UTILITY ROUTINE DISPATCHER
50	02	A4	3C	032F	540	MOVZWL	IHDR\$ACTIVOFF(R4),RO	:OFFSET TO ACTIVATION DATA (TRANSFER VECTORS
			0333	541		ADDL	R4,RO	:ADDRESS OF TRANSFER VECTOR ARRAY
50	54	CO	0333	541		PUSHL	(RO)	:ADDRESS OF TRANSFER VECTOR ARRAY
			0336	542		CALLS	#6,@(RO)+	:CALL IMAGE ENTRY POINT
90	06	FB	0338	543		JMP	G^EXE\$EXIT_IMAGE	
00000000'GF	17	033B	544	110\$:		.DSABL	LSB	
			0341	545				

```

0341 547 .SBTTL DCL$FORCEXIT - FORCE IMAGE EXIT
0341 548 :+
0341 549 : FORCE IMAGE EXIT IF ONE IS ACTIVE
0341 550 :
0341 551 : INPUTS:
0341 552 :
0341 553 : R2 = POSSIBLE ADDRESS OF ITEM ON STACK
0341 554 :
0341 555 : OUTPUTS:
0341 556 :
0341 557 : R2 = RELOCATED ADDRESS IF STACK IS SHUFFLED
0341 558 : R0 ALTERED, R1, R3 PRESERVED
0341 559 :-
0341 560
0341 561 DCL$FORCEXIT::
50 5E D0 0341 562 MOVL SP,R0
OF BB 0344 563 PUSHF #*M<R0,R1,R2,R3>
FCB7' 30 0346 564 BSBW DCL$RUNDOWN
OF BA 0349 565 POPR #*M<R0,R1,R2,R3>
5E 50 D1 034B 566 CMPL R0,SP
0B 13 034E 567 BEQL 10$
50 52 D1 0350 568 CMPL R2,R0
06 1F 0353 569 BLSSU 10$
52 50 C2 0355 570 SUBL R0,R2
52 5E C0 0358 571 ADDL SP,R2
05 05 035B 572 10$: RSB

```

```

:FORCE IMAGE EXIT
:SAVE SP ADR IN ORDER TO DETECT A SHUFFLE
:SAVE NAME STRING PARAMETERS
:RUN DOWN PREVIOUS IMAGE AND INDIRECT LEVELS
:RESTORE NAME STRING PARAMETERS
:WAS THE STACK SHUFFLED?
:BRANCH IF NOT
:WAS NAME STRING ON THE STACK?
:BRANCH IF NOT
:ADJUST THE ADDRESS
:TO REFLECT THE SHUFFLE

```

```

035C 574 .SBTTL ALLOCATE BUFFER AND BUILD DESCRIPTOR
035C 575
035C 576 :+ DCL$ALLOCBUF - ALLOCATE BUFFER AND BUILD DESCRIPTOR
035C 577
035C 578 THIS ROUTINE IS CALLED TO ALLOCATE A MESSAGE BUFFER ON THE STACK AND BUILD
035C 579 A BUFFER DESCRIPTOR FOR THE BUFFER.
035C 580
035C 581 INPUTS:
035C 582
035C 583 NONE.
035C 584
035C 585 OUTPUTS:
035C 586
035C 587 A MESSAGE BUFFER IS ALLOCATED ON THE STACK.
035C 588
035C 589 R2 = ADDRESS OF BUFFER DESCRIPTOR.
035C 590
035C 591 REGISTERS R3, R4, AND R5 ARE PRESERVED ACROSS CALL.
035C 592 :-
035C 593
035C 594 DCL$ALLOCBUF::
035C 595 POPR #^M<R0> ;ALLOCATE BUFFER AND BUILD DESCRIPTOR
035E 596 MOVAB -WRK_C_MSGBUFSIZ(SP),SP ;SAVE RETURN ADDRESS
0363 597 PUSHAB (SP) ;ALLOCATE SPACE TO STORE FORMAT STRING
0365 598 MOVZBL #WRK_C_MSGBUFSIZ,-(SP) ;BUILD OUTPUT BUFFER DESCRIPTOR
0369 599 MOVL SP,R2 ;COPY ADDRESS OF OUTPUT BUFFER DESCRIPTOR
036C 600 JMP (R0)

```



```

036E 602 .SBTTL COMMAND INTERPRETER EXIT HANDLER
036E 603
036E 604 :+ DCL$EXITHAND - COMMAND INTERPRETER EXIT HANDLER
036E 605
036E 606 THIS ROUTINE IS ENTERED WHEN A PREVIOUSLY INITIATED IMAGE EXITS. ITS FUNC-
036E 607 TION IS TO CLEAN UP THE STACK, SHUTDOWN THE IMAGE, AND RETURN CONTROL TO THE
036E 608 ADDRESS SPECIFIED BY THE TOP LONGWORD OF THE STACK.
036E 609
036E 610 INPUTS:
036E 611
036E 612 @4(AP) = REASON FOR EXIT.
036E 613
036E 614 OUTPUTS:
036E 615
036E 616 THE SAVED FRAME POINTER IS RESTORED, THE STACK IS CLEANED UP, IMAGE
036E 617 SHUT DOWN IS PERFORMED, AND CONTROL IS RETURNED TO THE ADDRESS SPECIFIED
036E 618 BY THE TOP LONGWORD OF THE STACK.
036E 619
036E 620 RO = REASON FOR EXIT.
036E 621 :-
036E 622
0000 036E 623 .ENTRY DCL$EXITHAND,*M<>
0370 624
0370 625 BSBW CLISGET PRC ;GET ADDRESS OF CLI WORK AREA
68 AB 08 AA 0373 626 BICW #PRC_M_EXIT,PRC_W_FLAGS(R11) ;CLEAR EXIT HANDLER ESTABLISHED
5D 04 AB D0 0377 627 MOVL PRC_C_SAVFP(R11),FP ;RESTORE SAVED FRAME POINTER
5A 5D D0 037B 628 MOVL FP,R10 ;AND RESTORE WRK ADDRESS
037E 629
037E 630 : ZERO COMMAND INTERPRETER COMMAND POINTERS
037E 631
00000000'GF D4 037E 632 CLRL G*CTL$GL_CLINTOWN ;ZERO CLINT OWN STORAGE POINTER
00000000'GF D4 0384 633 CLRL G*CTL$GL_DCLPRSOWN ;ZERO DCL PARSE OWN STORAGE
038A 634
038A 635 : ISSUE ERROR MESSAGE (IF ANY) RETURNED BY IMAGE IN RO
038A 636
50 04 BC D0 038A 637 10$: MOVL @4(AP),RO ;RETRIEVE FINAL EXIT STATUS
OC 50 E8 038E 638 BLBS RO,20$ ;BRANCH IF SUCCESSFUL
00000000'8F 50 D1 0391 639 CMPL RO,#SS$ _CLIFRCXT ;NEVER ISSUE CLI FORCED EXIT MESSAGE
03 13 0398 640 BEQL 20$ ;IF IMPLIED IMAGE RUNDOWN
FC63' 30 039A 641 BSBW DCL$ERRORMSG ;ISSUE ERROR MESSAGE USING PER-IMAGE
039D 642 ;MESSAGES IF PRESENT
039D 643
039D 644 : RUNDOWN ALL RMS FILES AND FLUSH ANY DATA RECORDS
039D 645
FC60' 30 039D 646 20$: BSBW DCL$SHUTDOWN ;SHUT DOWN IMAGE
52 D5 03A0 647 TSTL R2 ;ANY DATA RECORDS SKIPPED?
0A 13 03A2 648 BEQL 30$ ;IF EQL NO
FC52' 30 03A4 649 STATUS SKPDAT ;SET SKIPPED DATA STATUS
03AB 650 BSBW DCL$ERRORMSG ;OUTPUT ERROR MESSAGE
03AE 651
03AE 652 : RUNDOWN THE IMAGE
03AE 653
00AF CB 18 8A 03AE 654 30$: $RUNDWN_S #PSL$C USER ;RUN DOWN THE IMAGE
03B7 655 BICB #<PRC_M_EXEONLY ! PRC_M_PRIV>,PRC_B_FLAGS2(R11) ;RESET "EXECUTE ONLY" AND
03BC 656 ;"PRIVILEGED" IMAGE BITS
03BC 657
03BC 658 CLRBIT PRC_V_IRUNDWN,PRC_B_IMGFLAG(R11) ;INDICATE IMAGE RUNDOWN.

```

```

03C0 659 :
03C0 660 : RESTORE STACK POINTER TO SP SAVED ON THE ACTIVATE CALL
03C0 661 :
SE F4 AA D0 03C0 662 : MOVL WRK_L_SAVSP(R10),SP ;RESTORE SAVED STACK POINTER
03C4 663 :
03C4 664 : SET THE FINAL RETURN STATUS IN $STATUS AND TAKE ANY ON CONDITION
03C4 665 :
50 04 BC D0 03C4 666 : MOVL @4(AP),R0 ;GET FINAL IMAGE STATUS
09 50 E8 03C8 667 : BLBS R0,40$ ;BRANCH IF SUCCESSFUL
00000000'BF 50 D1 03C8 668 : CMPL R0,$SS$_CLIFRCXT ;NEVER SET IMPLIED IMAGE RUNDOWN STATUS
03 13 03D2 669 : BEQL 50$ ;BRANCH IF FORCED EXIT
FC29' 30 03D4 670 40$: BSBW DCL$SET_STATUS ;SET $STATUS AND TAKE ON CONDITION
03D7 671 : ;NOTE: WILL NOT RETURN IF ON CONDITION
03D7 672 :
03D7 673 : SET STATUS TO SUCCESS AND SET NOSTAT BIT TO INDICATE STATUS ALREADY SET
03D7 674 :
50 01 D0 03D7 675 50$: SETBIT #WRK_V_NOSTAT,WRK_W_FLAGS(R10) ;INDICATE STATUS ALREADY SAVED
05 03DC 676 : MOVL #1,R0 ;AND SET SUCCESSFUL
03DF 677 : RSB ;RETURN TO CALLER (COMMAND OR RUNDOWN)
03E0 678 :
03E0 679 : .END

```

IMAGEXECT  
Symbol table

- IMAGE EXECUTION

F 14

15-SEP-1984 23:54:20 VAX/VMS Macro V04-00  
4-SEP-1984 23:41:04 [DCL.SRC]IMAGEXECT.MAR;1

Page 17  
(11)

SS.TMP1	= 00000001		
SS.TMP2	= 00000060		
SS11	= 00000000		
SINTIMAGES	= 00000086		
CLISGET_PRC	*****	X	02
CLISM_BATCH	= 00000008		
CLISM_INDIRECT	= 00000010		
CLISV_BATCH	= 00000003		
CLISV_DBGTRU	= 00000001		
CLISV_DEBUG	= 00000000		
CLISV_INDIRECT	= 00000004		
CLISV_VERIFY	= 00000002		
CLISV_VFYINP	= 00000005		
CLIS_IMAGEFNF	= 00038882		
CLIS_IMGNAME	= 000388AA		
CLIS_NOTFR	= 0003889A		
CLIS_SKPDAT	= 00038120		
CTLSAG_CLIDATA	*****	X	02
CTLSGL_CLINTOWN	*****	X	02
CTLSGL_DCLPRDOWN	*****	X	02
DCL\$ALCOBUF	0000035C	RG	02
DCL\$CLOSE_PPFS	00000073	RG	02
DCL\$ERRORMSG	*****	X	02
DCL\$EXITHAND	0000036E	RG	02
DCL\$EXTIMAGE	000000E9	RG	02
DCL\$FORCEEXIT	00000341	RG	02
DCL\$GETDVAL	*****	X	02
DCL\$HIGH_LIMIT	00000297	RG	02
DCL\$LOGOUT	0000005A	RG	02
DCL\$LOW_LIMIT	00000288	RG	02
DCL\$MCR	000000A8	RG	02
DCL\$PUTMSG	*****	X	02
DCL\$STORE_OUTPUT	*****	X	02
DCL\$RUN	0000002B	RG	02
DCL\$RUNDOWN	*****	X	02
DCL\$SCNTRY	*****	X	02
DCL\$SET_STATUS	*****	X	02
DCL\$SHUTDOWN	*****	X	02
DCL\$UNSTACK	*****	X	02
DCL\$UTLSERV	*****	X	02
EXE\$CATCH_ALL	*****	X	02
EXE\$EXIT_IMAGE	*****	X	02
EXTDEFAULT	00000000	R	02
FAB\$B_FNS	= 00000034		
FAB\$B_FNA	= 0000002C		
FAB\$B_NAM	= 00000028		
FAB\$B_STV	= 0000000C		
FAB\$W_IFI	= 00000002		
FORCEEXIT	000000FF	R	02
IFDSV_EXEONLY	= 00000000		
IFDSV_PRIV	= 00000001		
IFDSW_FLAGS	= 00000010		
IND\$B_LNKFLAGS	= 00000020		
IND\$B_SYSVER	= 00000028		
IND\$B_LNKNOTFR	= 00000001		
IND\$W_ACTIVOFF	= 00000002		
IMG_K_ALLOCATE	= 00000080		

IMG_K_ASSIGN	= 00000081		
IMG_K_ATTACH	= 000000A9		
IMG_K_CANCEL	= 000000AB		
IMG_K_CLOSE	= 00000082		
IMG_K_CONNECT	= 000000B4		
IMG_K_CONTINUE	= 00000083		
IMG_K_CRETABLE	= 000000B3		
IMG_K_DEALLOCAT	= 00000084		
IMG_K_DEASSIGN	= 00000085		
IMG_K_DEBUG	= 00000086		
IMG_K_DECK	= 00000087		
IMG_K_DEFINE	= 00000088		
IMG_K_DEFKEY	= 000000AE		
IMG_K_DELKEY	= 000000B0		
IMG_K_DELSYM	= 0000008A		
IMG_K_DEPOSIT	= 00000089		
IMG_K_DISCONNECT	= 00000085		
IMG_K_EOD	= 0000008B		
IMG_K_EOJ	= 000000AA		
IMG_K_EXAMINE	= 0000008C		
IMG_K_EXIT	= 0000008D		
IMG_K_EXTIMAGE	= 0000008E		
IMG_K_GOTO	= 0000008F		
IMG_K_IF	= 00000090		
IMG_K_INQUIRE	= 00000091		
IMG_K_LOGOUT	= 00000092		
IMG_K_MCR	= 00000093		
IMG_K_ON	= 00000094		
IMG_K_OPEN	= 00000095		
IMG_K_READ	= 00000096		
IMG_K_RECALL	= 000000AD		
IMG_K_RUN	= 00000097		
IMG_K_SETCTLY	= 00000098		
IMG_K_SETDEFAULT	= 00000099		
IMG_K_SETFLUSH	= 000000B2		
IMG_K_SETKEY	= 000000B1		
IMG_K_SETON	= 0000009A		
IMG_K_SETPROMPT	= 000000AC		
IMG_K_SETPROT	= 0000009B		
IMG_K_SETUIC	= 0000009C		
IMG_K_SETVERIFY	= 0000009D		
IMG_K_SHOWDEF	= 0000009E		
IMG_K_SHOWKEY	= 000000AF		
IMG_K_SHOWPROT	= 0000009F		
IMG_K_SHOWQUOTA	= 000000A0		
IMG_K_SHOWSTAT	= 000000A1		
IMG_K_SHOWSYMBL	= 000000A2		
IMG_K_SHOWTIME	= 000000A3		
IMG_K_SHOWTRAN	= 000000A4		
IMG_K_SPAWN	= 000000A8		
IMG_K_STDP	= 000000A5		
IMG_K_WAIT	= 000000A6		
IMG_K_WRITE	= 000000A7		
INITIATE	00000102	R	02
LOGOUTIMG	00000015	R	02
MCRIMG	0000001E	R	02
MMG\$IMGHDRBUF	*****	X	02



IMAGEXECT  
Symbol table

- IMAGE EXECUTION

G 14

15-SEP-1984 23:54:20 VAX/VMS Macro V04-00  
4-SEP-1984 23:41:04 [DCL.SRC]IMAGEXECT.MAR;1

Page 18  
(11)

NAMSB_ESL	= 00000008
NAMSB_RSL	= 00000003
NAMSL_RSA	= 00000004
PPDSB_NPROCS	0000001C
PPDSC_LENGTH	00000168
PPDSK_LENGTH	00000168
PPDSL_INPDEV	00000044
PPDSL_LGI	00000014
PPDSL_LSTSTATUS	00000018
PPDSL_OUTDEV	00000064
PPDSL_PRC	00000008
PPDSQ_CLIREG	00000004
PPDSQ_CLISYMTBL	0000000C
PPDST_FILENAME	00000068
PPDST_INPDVI	00000028
PPDST_OUTDVI	00000048
PPDSW_FLAGS	00000002
PPDSW_INPCHAN	0000001E
PPDSW_INPDID	0000003E
PPDSW_INPFID	00000038
PPDSW_INPFI	00000020
PPDSW_INPISI	00000022
PPDSW_OUTDID	0000005E
PPDSW_OUTFID	00000058
PPDSW_OUTIFI	00000024
PPDSW_OUTISI	00000026
PPDSW_SIZE	00000000
PRC_B_CONTINUE	000000F3
PRC_B_DEFRADIX	000000AE
PRC_B_EXMDEPMOD	000000AD
PRC_B_EXMDEPWID	000000AC
PRC_B_EXONLYL	0000012D
PRC_B_FLAGS2	000000AF
PRC_B_IMGFLAG	00000078
PRC_B_OUTFLAGS	0000012C
PRC_B_PROMPTLEN	000000F0
PRC_C_LENGTH	00000534
PRC_G_COMMANDS	00000133
PRC_G_PROMPT	000000F4
PRC_K_LENGTH	00000534
PRC_L_CURRKEY	00000048
PRC_L_EXMDEPADR	000000A8
PRC_L_EXTARG	00000094
PRC_L_EXTBLK	0000008C
PRC_L_EXTCOD	0000009C
PRC_L_EXTHND	00000090
PRC_L_EXTPRM	00000098
PRC_L_IDFLNK	000000BC
PRC_L_IMGACTSTS	00000080
PRC_L_INDCLOCK	0000007C
PRC_L_INDEPTH	0000005C
PRC_L_INDFAB	0000001C
PRC_L_INDINPRAB	00000014
PRC_L_INDOUTRAB	00000018
PRC_L_INPRAB	00000008
PRC_L_LASTKEY	0000004C
PRC_L_LSTSTATUS	000000B0

PRC_L_ONCTLY	000000B8
PRC_L_ONERROR	0000006C
PRC_L_OUTOFBAND	000000B4
PRC_L_OUTRAB	0000000C
PRC_L_OUTRABCTX	00000118
PRC_L_PPFLIST	00000070
PRC_L_RECALLPTR	0000012F
PRC_L_RESTART	00000058
PRC_L_SAVAP	00000000
PRC_L_SAVFP	00000004
PRC_L_SEVERITY	00000050
PRC_L_SPWN	000000C0
PRC_L_STACKLM	000000A4
PRC_L_STACKPT	000000A0
PRC_L_STATUS	00000054
PRC_L_STS	00000084
PRC_L_STV	00000088
PRC_L_SYMBOL	00000060
PRC_L_TMBX	00000074
PRC_L_TRLIST	00000010
PRC_M_DBGQUAL	= 00000200
PRC_M_DBGTRUE	= 00000400
PRC_M_EXEONLY	= 00000008
PRC_M_EXIT	= 00000008
PRC_M_PRIV	= 00000010
PRC_Q_ALLOCREG	00000020
PRC_Q_COMMAND	000000E0
PRC_Q_FLUSHTIME	000000D0
PRC_Q_GLOBAL	00000028
PRC_Q_IMAGENAME	000000D8
PRC_Q_KEYPAD	00000040
PRC_Q_LABEL	00000030
PRC_Q_LOCAL	00000038
PRC_Q_SAVEPRIV	000000E8
PRC_T_OUTDVI	0000011C
PRC_V_AUTOLOGO	= 00000008
PRC_V_CNTRL	= 00000001
PRC_V_DBGQUAL	= 00000009
PRC_V_DBGTRUE	= 0000000A
PRC_V_EXEONLY	= 00000003
PRC_V_EXIT	= 00000003
PRC_V_IRUNDWN	= 00000000
PRC_V_MODE	= 00000006
PRC_V_PRIV	= 00000004
PRC_V_RUNDEF	= 00000002
PRC_V_VERIFY	= 00000007
PRC_W_ASTIOSB	000000C6
PRC_W_ASTRETN	000000C8
PRC_W_ASTSTATUS	000000C4
PRC_W_ATTMBX	0000007A
PRC_W_FLAGS	00000068
PRC_W_INPCHAN	00000064
PRC_W_ONLEVEL	0000006A
PRC_W_OUTIFI	00000114
PRC_W_OUTISI	00000116
PRC_W_OUTMBXCHN	000000CA
PRC_W_OUTMBXREF	000000CE

IMAGEXECT  
Symbol table

- IMAGE EXECUTION

H 14

15-SEP-1984 23:54:20 VAX/VMS Macro V04-00  
4-SEP-1984 23:41:04 [DCL.SRC]IMAGEXECT.MAR;1

Page 19  
(11)

```

PRC_W_OUTMBXSIZ      000000CC
PRC_W_PMPTCTRL       000000F1
PRC_W_WAITIOSB       00000066
PSL$C_USER           = 00000003
PSL$V_PRVMOD         = 00000016
PTR_B_LEVEL          00000004
PTR_B_NUMBER         00000005
PTR_B_PARMCNT        00000006
PTR_B_VALUE          00000000
PTR_C_LENGTH         0000000C
PTR_K_ENDLINE        = 00000004
PTR_K_LENGTH         0000000C
PTR_K_PARAMETER      = 00000003
PTR_L_DESCR          00000000
PTR_L_ENTITY         00000008
RAB$L_CTX            = 00000018
RMS$ FNF             *****
RUNDEFAULT           00000010 R X 02
SETIMGDEF            000000E9 R R 02
SETIMGNAME           000000B5 R R 02
SETRUNDEF            000000FA R 02
SHR$ ACTIMAGE        *****
SILENT LOGOUT        *****
SS$ CLIFRCEXT        *****
SS$ NORMAL           *****
SS$ SYSVERDIF        *****
ST$M FAC_NO          = 0FFF0000
ST$S FAC_NO          = 0000000C
ST$V FAC_NO          = 00000010
ST$V INHTB_MSG       = 0000001C
SY$SCANEXH           *****
SY$CLOSE             *****
SY$DCLEXH            *****
SY$EXIT              *****
SY$IMGACT            *****
SY$IMGFIX            *****
SY$PUTMSG            *****
SY$RUNDN             *****
SY$SETEXV            *****
SYS$PRINT            00000022 R
SYS$PRTSIZ           = 00000009
WRK_B_CMDOPT         FFFFFFFC3
WRK_B_MAXPARM        FFFFFFFD0
WRK_B_MINPARM        FFFFFFFD1
WRK_B_PARMCNT        FFFFFFFCE
WRK_B_PARMSUM        FFFFFFFCF
WRK_B_RECALLCNT      FFFFFFFC5
WRK_B_VALLEV         FFFFFFFC4
WRK_B_VERBTYP        FFFFFFFC2
WRK_C_LENGTH         FFFFF486
WRK_C_MSGBUFSIZ      = 00000084
WRK_G_BUFFER         FFFFF492
WRK_G_INPBUF         FFFFF896
WRK_G_RESULT         FFFFF986
WRK_K_LENGTH         FFFFF486
WRK_L_CHARPTR        FFFFF48E
WRK_L_DISALLOW       FFFFFFE6

```

```

WRK_L_ERRORRTN      FFFFF9AE
WRK_L_EXPANDPTR     FFFFF486
WRK_L_IMAGE         FFFFFFFE2
WRK_L_MARKPTR       FFFFF48A
WRK_L_PAROUT        FFFFFFFD2
WRK_L_PMPTADDR      FFFFF9A2
WRK_L_PROMPTRTN     FFFFF9A6
WRK_L_PROPTR        FFFFFFFC6
WRK_L_QUABLK        FFFFFFFCA
WRK_L_READRTN       FFFFF9AA
WRK_L_RECALLPTR     FFFFFFFEA
WRK_L_RSLND         FFFFFFFB6
WRK_L_RSLNXT        FFFFFFFBA
WRK_L_SAVAP         FFFFFFFF8
WRK_L_SAVFP         FFFFFFFFC
WRK_L_SAVSP         FFFFFFFF4
WRK_L_SIGNALRTN     FFFFFFFD6
WRK_L_SPECRTN       FFFFF9B2
WRK_L_TAB_VEC       FFFFFFFDE
WRK_L_VERB          FFFFFFFBE
WRK_V_NOSTAT        = 00000008
WRK_W_FLAGS         FFFFFFFF0
WRK_W_FLAGS2        FFFFFFFF2
WRK_W_IMGCHAN       FFFFFFFEE
WRK_W_PMPTLEN       FFFFF99E
_$$_                = 00000000

```



-----  
! Psect synopsis !  
-----

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR
\$ABSS	FFFFFFFFC ( 0.)	01 ( 1.)	NOPIC USR
DCL\$ZCODE	000003E0 ( 992.)	02 ( 2.)	NOPIC USR

-----  
! Performance indicators !  
-----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	15	00:00:00.07	00:00:01.03
Command processing	102	00:00:00.64	00:00:05.73
Pass 1	384	00:00:15.47	00:01:03.55
Symbol table sort	4	00:00:02.12	00:00:06.27
Pass 2	120	00:00:03.03	00:00:12.20
Symbol table output	35	00:00:00.25	00:00:01.40
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	662	00:00:21.60	00:01:30.20

The working set limit was 1500 pages.  
82068 bytes (161 pages) of virtual memory were used to buffer the intermediate code.  
There were 80 pages of symbol table space allocated to hold 1390 non-local and 42 local symbols.  
679 source lines were read in Pass 1, producing 19 object records in Pass 2.  
58 pages of virtual memory were used to define 42 macros.

-----  
! Macro library statistics !  
-----

Macro library name	Macros defined
-\$255\$DUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
-\$255\$DUA28:[DCL.OBJ]DCL.MLB;1	10
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	2
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	23
TOTALS (all libraries)	35

1665 GETS were required to define 35 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:IMAGEXECT/OBJ=OBJ\$:IMAGEXECT MSRC\$:IMAGEXECT/UPDATE=(ENH\$:IMAGEXECT)+EXECML\$/LIB+LIB\$:DCL/LIB+SYSS\$LIBRARY:SYSBLDMLB/L



0070

AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY